DEPARTMENT OF DEFENCE
EFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Implementation of Call Activity Detection for ATM Voice Services in Project Parakeet

W.D. Blair and A.B. Reynolds

DSTO-TR-1149

Implementation of Call Activity Detection for ATM Voice Services in Project Parakeet

W.D. Blair and A.B. Reynolds

Communications Division
Electronics and Surveillance Research Laboratory

DSTO-TR-1149

ABSTRACT

Asynchronous Transfer Mode (ATM) has been introduced into the land tactical communications system (specifically the Parakeet system) and offers the potential for increasing the effectiveness of the communications trunks by dynamically sharing the capacity between competing demands. Such a capability has been limited in respect of the trunks between the Parakeet circuit switches since Parakeet uses military standard protocols that do not integrate easily with the civil standard ATM. The Parakeet Adaptive Rate ATM Trunk (PARAT) was conceived to overcome this interface problem. The PARAT offers substantially improved data communications performance where lulls in voice usage can be exploited by data services sharing the ATM link. This report describes PARAT, in particular the mechanism to identify the moment-to-moment capacity requirements of the Parakeet inter-switch trunk and the method of carrying the variable bit rate stream over ATM.

RELEASE LIMITATION

Approved for public release

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION DSTO

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-011-861
March 2001

APPROVED FOR PUBLIC RELEASE

Implementation of Call Activity Detection for ATM Voice Services in Project Parakeet

Executive Summary

The Australian Defence Force is currently procuring, under Project Parakeet, a new digital communications network for use by forces deployed in the battlefield. It is based on a military standard time division multiplexed (TDM) system known as Eurocom. As part of the current phase of Parakeet, a limited Asynchronous Transfer Mode (ATM) capability is being explored with the aim of better integrating the voice and data uses of the network. The unsophisticated transmission of voice over ATM as a Constant Bit Rate (CBR) stream incurs bandwidth penalties because of the addition of overheads but this is the approach taken by the Parakeet Interim ATM Hub Assembly.

A major strength of ATM is its ability to carry variable bit rate (VBR) traffic and the impact of the overheads associated with the ATM protocol can be eliminated if the voice traffic adopts a VBR approach. There are two approaches:

- at the trunk level the capacity requirements can be varied to reflect the number of calls being connected at each moment (call activity); and
- at the call level the capacity requirement of each channel can vary according to the on-off activity in the conversation, i.e. speech activity (silence detection).

This report describes in detail the mechanism for implementing a Eurocom trunk VBR service over ATM and the concept demonstrator (Parakeet Adaptive Rate ATM Trunk - PARAT) that has been built based on these ideas. The Concept Demonstrator could perhaps become the basis of an initial operational capability, but a more scalable solution may require further development by industry.

By implementing this VBR approach, substantially improved performance can be offered to the data services during lulls in voice usage at no extra cost in trunk bearer capacity.

Authors

W.D. Blair
Communications Division

Bill Blair is a Senior Research Scientist in the Network Architecture Group of the Defence Science and Technology Organisation's (DSTO) Communications Division having joined DSTO from the Australian Army. He is involved in research into networking techniques to control Quality of Service (especially Asynchronous Transfer Mode - ATM) for military strategic and tactical communications.

A.B. Reynolds
Communications Division

Alfred Reynolds is a Research Engineer in Network Architecture Group of the Defence Science and Technology Organisation's (DSTO) Communications Division having joined DSTO after completing a Systems Engineering Degree at the Australian National University. He is involved in research into tactical communication systems.

Contents

1. Introduction

The Australian Defence Force is currently procuring, under Project Parakeet, a new digital communications network for use by forces deployed to the battlefield. It is based on a military standard time division multiplexed (TDM) system known as EUROCOM [1], which is similar in concept to the civilian integrated services digital network (ISDN). A Parakeet deployment would see a number of circuit switches interconnected in a mesh by radio or satellite links (trunks) each typically carrying 30 TDM digital user channels. Voice systems are a fundamental element of a military tactical communications system, and comprise the central element of the Parakeet network; nevertheless, increasing data communications demands have pushed for increased data capacity in the network. As part of the current phase of Parakeet, a limited Asynchronous Transfer Mode (ATM) capability is being explored with the aim of better integrating the voice and data uses of the network.

There are definite network management benefits from integrating voice services into the ATM network. Nevertheless, the unsophisticated transmission of voice over ATM incurs bandwidth penalties because of the addition of ATM headers and other overheads in the Constant Bit Rate (CBR) digital voice stream. This occurs in the approach taken by the Parakeet Interim ATM Hub Assembly fielded in association with the C-Band upgrade of the Parakeet Satellite Assemblage. A major strength of ATM is its ability to carry variable bit rate (VBR) traffic and as described in a previous DSTO report [2], the impact of the overheads associated with the ATM protocol can be eliminated if the voice traffic employs a VBR approach. Indeed by implementing this VBR approach, substantially improved performance can be experienced by the data services sharing the ATM intersite link during lulls in voice usage. This comes at no extra cost in bearer capacity. There are two approaches:

- at the trunk level the capacity usage can be varied to reflect the number of calls being connected at each moment (call activity); and
- at the call level the capacity usage of each channel can vary according to the on-off activity in the conversation, i.e. speech activity (silence detection).

2. Aim

This report describes in detail the mechanism for implementing a VBR service to carry a EUROCOM trunk over ATM. The description also covers the concept demonstrator (Parakeet Adaptive Rate ATM Trunk - PARAT) that has been built based on these ideas.

3. Overall Design

There are three fundamental operations that comprise a VBR adaptor:

- Demultiplexing of the EUROCOM TDM trunk and identification of active channels.
- Creation of a frame (or protocol data unit – PDU) of voice data to carry the information from only the active channels – inactive channel information is not carried (inherently this frame will be variable in length)
- Segmentation of the frame into ATM cells. This is done through an ATM Adaptation Layer (AAL). Of the four standard AALs there are only two that are relevant: AAL1 that was intended for real time traffic and AAL2 previously known as AAL-CU.

At the receiving end of the VBR trunk an inverse process is required to restore the TDM trunk for connection into the circuit switch.

3.1 Active Channel Identification

The EUROCOM trunk channels are bit interleaved ie one bit per channel in each TDM time slot. There are 32 time slots when the trunk is operating at 512 kbps and 16 time slots for 256 kbps. Specific multiplexer channels can be identified since the first time slot carries a repeating synchronisation pattern. The next time slot is a common channel signalling stream while the remainder of the time slots carry separate traffic channels. At any particular moment, each traffic channel can be classified as active (committed to a trunk connection ie an intersite telephone call) or idle. Testing revealed that the bit pattern transmitted on idle channels is a continuous "101010" etc whereas active channels carry a digital stream representing the audio signal.

This is discussed in more detail in Appendix A.

3.2 Variable Bit Rate Frame

As discussed in [2] there are at least three mechanisms for carrying voice in variable bit rate format: an individual CBR AAL1 virtual circuit for each active channel, AAL2 and Dynamic Bandwidth Circuit Emulation Service (DBCES). DBCES, as defined in [3], is the recommended approach to form the basis for this application. It has better latency characteristics than the use of an AAL1 virtual circuit per channel and better bandwidth efficiencies than AAL2. The PARAT implementation requires only a slight change in interpretation to the commercial standard to account for the differences between the military standard TDM structure and the civil standard.

A DBCES frame comprises two elements:

- a header acting as a bit mask identifying which of the time slots in the TDM stream are active and thus which channels have audio data contained in the frame; and

- a stream of data from the active channels only. In the PARAT demonstrator, for each DBCES frame there are 22 bytes of data from each active channel. All 22 bytes must exhibit a continuous idle pattern for the channel to be considered inactive.

The frame is carried using the Structured Data Transfer (SDT) mode of AAL1 [4] and the standard dictates limits to the minimum frame size that can be carried. Nevertheless, this constraint assists in limiting end to end latency imposed by the PARAT. The frame structure and the latency implications are discussed in more detail in Appendix B.

3.3 ATM Encapsulation

The frame is carried in the SDT mode of AAL1 strictly in accordance with the civil standards [4]. The process of fragmenting the frame into AAL1 cells is shown diagrammatically in Figure 1. More detail is available in Appendix B.

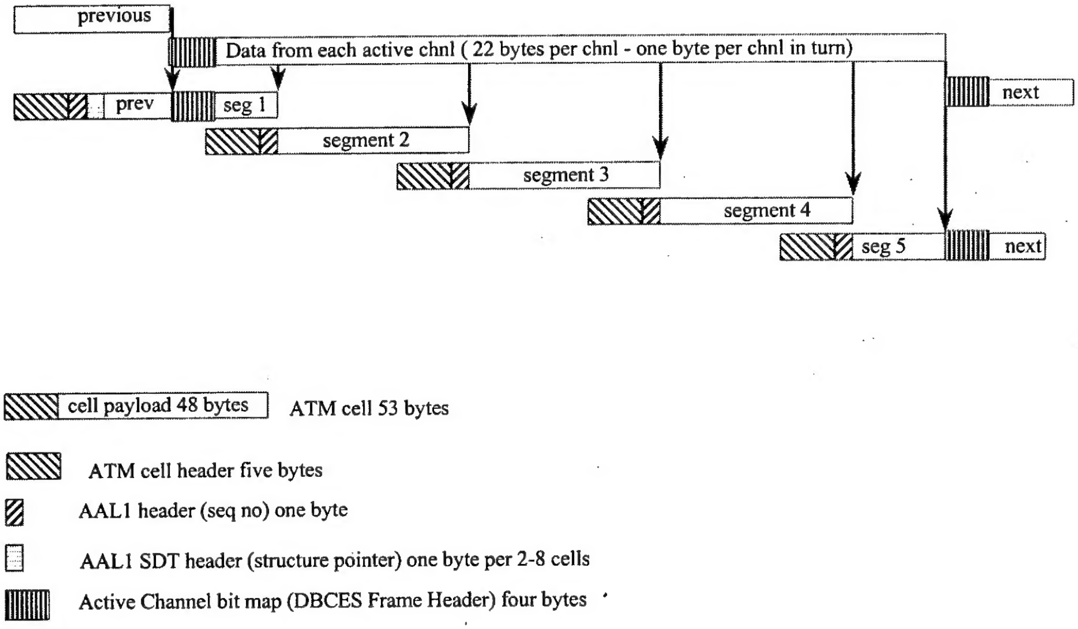
previous
Data from each active chnl (22 bytes per chnl - one byte per chnl in turn)
prev
seg 1
segment 2
segment 3
segment 4
seg 5
next
next
cell payload 48 bytes
ATM cell 53 bytes
ATM cell header five bytes
AAL1 header (seq no) one byte
AAL1 SDT header (structure pointer) one byte per 2-8 cells
Active Channel bit map (DBCES Frame Header) four bytes

Figure 1. Fragmenting DBCES Frames into ATM Cells.

4. PARAT Implementation

4.1 Bandwidth Usage

As discussed in Appendix B a minimum of five channels is sent over the VBR connection (even if those channels are carrying the idle pattern). This leads to a minimum bit rate consumed by the PARAT system of 94 kbps. The bit rate for the maximum 30 traffic channels (assuming both synchronisation and signalling channels are also sent) is 582 kbps. See Chart 1 for the bit rate consumption of the DBCES as a function of the number of active traffic channel compared with the current AAL1 based approach for 512 kbps operation that uses a constant 577 kbps. The chart assumes both synchronisation and signalling channels are sent over the DBCES.

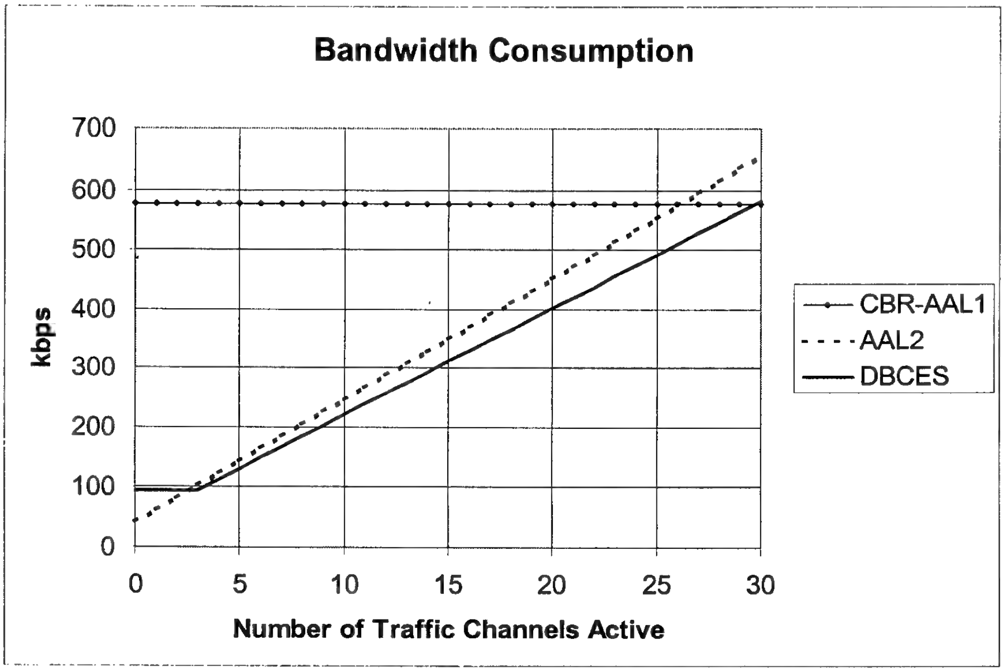

Chart 1. Bandwidth Consumption as a Function of Number of Active Channels

The bandwidth chart is different to that provided in [2]. The PARAT DBCES implementation is slightly less efficient than the earlier proposal most notably because:

- PARAT figures assume both the synchronisation channel and the signalling channel will always be included in the DBCES frame whereas [2] assumed only signalling.
- During implementation of the PARAT, the issue of minimum frame sizes was revealed.

The revised DBCES bandwidth characteristics were analysed using the traffic model developed in [2] and this revealed bandwidth savings remain similar to those previously reported. The traffic model developed in [2] was based on traffic statistics provided in the Parakeet specification [5] and determined that the peak requirement during the busiest period on the busiest trunk would see 23 channels active for traffic. Table 1 provides the full comparison of the PARAT with the original TDM stream and the CBR ATM stream as carried in the current ATM implementation.

	PARAT DBCES (kbps)	**Savings compared with 512kbps TDM**	**Savings compared with 512kbps over AAL1 (577kbps)**
Peak requirement of 23 active channels	456	11%	21%
Weighted Average B/W Requirement over the Busy Hour	250	51%	57%

Table 1. Bandwidth Savings Resulting from Call Activity.

4.2 PARAT Concept Demonstrator as an Adjunct Processor

The PARAT has been implemented as an adjunct processor. It is hosted on a standard PC workstation with an ATM network card connected to the Parakeet ATM switch as shown in Figure 2. For simplicity the description below refers to the connection from circuit switch to the bearer, but the system is full duplex and acts in both directions.

The Parakeet ATM switch provides the physical interface to the circuit switch via a EUROCOM to Digital Adaptor (EDA) a device to handle physical layer differences between the EUROCOM circuit switch connection and the commercial standard ATM switch ports. The Parakeet ATM switch encapsulates the synchronous stream into CBR AAL1 and directs the stream to the PARAT over a commercial standard OC-3 ATM link. This OC-3 link is also used to carry the VBR DBCES stream back to the Parakeet ATM switch so it can be combined with other ATM streams (such as from the IP router) and directed over the intersite link.

Since the PARAT receives the TDM trunk via a CBR AAL1 connection to the ATM switch, the adjunct processor architecture has thus added a fourth operation to the three described in the Overall Design - the demonstrator has to extract the TDM stream out of the AAL1 cell stream.

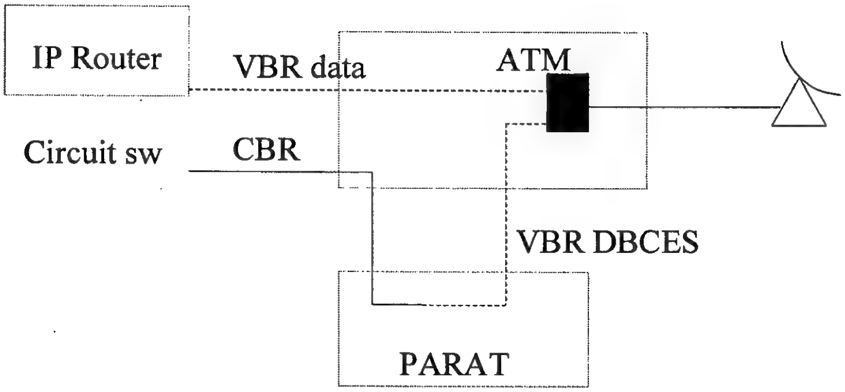

Figure 2. PARAT Connections

4.3 PARAT Software

4.3.1 General

In carrying out its operation the PARAT is fundamentally processing four streams of data shown diagrammatically in Figure 3:

- One input constant bit rate stream from the circuit switch
- One output variable bit rate stream to the distant PARAT
- One input variable bit rate stream from the distant PARAT
- One output constant bit rate stream to the circuit switch

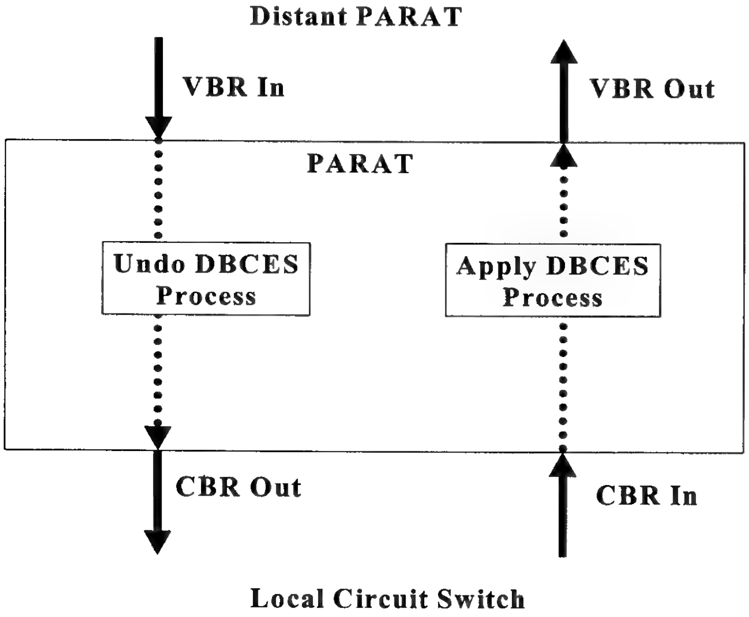

Figure 3. PARAT Streams

Since all four streams are carried on a single physical interface of the concept demonstrator, a single instance of network interface driver software is employed. It is convenient to aggregate into one code module the process of sending and receiving of VBR frames via SDT AAL1 and place the sending and receiving of the CBR AAL1 stream in another. The application of the DBCES process occurs in the main code module. This structure is shown in Figure 4. The outline software design is at Appendix C while the detailed code is available at http://web-fhp.dsto.defence.gov.au/na-group/landarch/parat/ (only accessible via the Australian Department of Defence intranet) or on request.

The DBCES processes operate on discrete frames (or time periods) of channel data. The length of the frame is a compromise between the latencies introduced by collecting and processing frames of traffic that increase with larger frames; competing against bit rate efficiencies that improve with larger frames (through greater amortisation of protocol overheads). For the concept demonstrator, a frame of 11 ms (ie 22 bytes per channel) was selected.

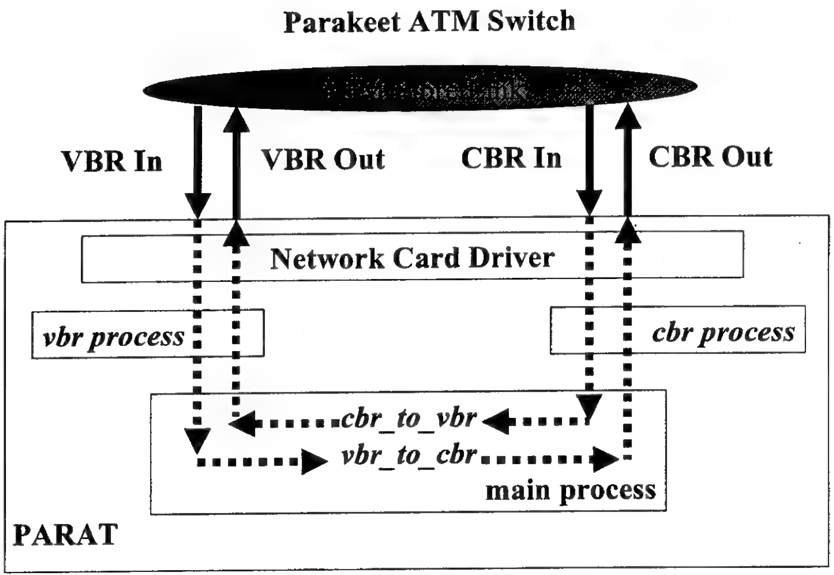

Figure 4. PARAT Main Software Modules

4.3.2 Timing Issues

The CBR AAL1 stream coming into the *cbr process* is passing through a single ATM switch and a high speed fibre optic link so cells arrive at regular intervals unaffected by the ATM network. The *cbr process* aggregates a package of data equivalent to 11 ms of the synchronous stream - effectively a CBR 'frame' - and presents this to the *cbr_to_vbr process* for conversion into a DBCES frame. This will not occur at perfectly regular intervals since the frame is not an integral number of CBR AAL1 payloads. For instance, for the 32 channel (512 kbps) case the CBR 'frame' at 32x22=704 bytes is equivalent to 14.98 CBR AAL1 payloads of 47 bytes. While frames will be available on average every 11 ms, they will mostly arrive at 11.02 ms (the time to collect 15 cells) and every 517 ms one arrives at 10.28 ms (14 cells).

At these slightly irregular intervals, the *cbr_to_vbr process* is initiated to undertake the DBCES conversion. The *vbr process* is then initiated to process the DBCES frame into ATM cells and pass these down to the network driver. Since DBCES frames are unlikely to be an integral number of SDT AAL1 payloads there will likely be a partially filled cell that cannot be sent until it is completed. The receipt of the next DBCES frame into the sending *vbr process* will complete any partially filled cell of the previous frame, but will likely itself leave a partially filled cell.

At the distant PARAT, the *vbr process* would normally see the last portions of data for a DBCES frame at the time of receipt of the start of the next frame ie at fairly regular intervals driven by the receipt of CBR 'frames' at the source PARAT. For about 2% of DBCES frames, the last byte of the frame will fill the last byte of a cell. In this case, the *vbr_to_cbr process* will be initiated early - the degree of 'earliness' will be especially apparent if the frame is small (few channels active). Timing of the initiation of the *vbr_to_cbr process* will also be disturbed by a limited amount of ATM network jitter. Jitter is variations in the end to end delay across the network and will be limited in this case since the PARAT connection will be given the highest priority and should travel through two ATM switches only: local switch and distant switch. Careful buffer management is required to handle the jitter while discerning between jittered frame arrival and lost frames. Frames can be lost through ATM cells being discarded because of congestion (most probably at the sending ATM switch) or errors in the SDT mechanism.

Regardless of the cause of upstream jitter, the output *cbr process* must stream CBR AAL1 cells to the local ATM switch for reassembly and transmission via the EDA to the circuit switch. Since the connection to the circuit switch is a synchronous, real time stream, the local ATM switch will have only a small dejitter buffer and the *cbr process* must ensure that this does not underflow. The *cbr process* along with traffic shaping in the PARAT ATM network card must ensure regular arrival of the data at the output port of the Parakeet ATM switch.

The standard clock available to the application software on the PC has an accuracy of only +/- 10 ms. This clock is certainly not sufficient resolvable to rate control the sending of ATM cells in the CBR streams. There is however another source of accurate clocking. At the original source of the TDM stream, the Parakeet circuit switch, accurate hardware based clock is available. This is employed to transmit an accurately clocked synchronous bit stream to the ATM switch. This bit stream is segmented into AAL1 ATM cells and these arrive from the network into the *cbr process* at regular intervals - approximately 0.7ms for the 32 channel/512 kbps case. This constant arrival of cells on the CBR stream could be used to rate control the sending of ATM cells back to the ATM switch. As it happens this is not needed for the demonstrator as the network interface card in the PARAT PC offers an adequate traffic shaping service.

Since the software clock resolution is of the same order as the length of time carried by the DBCES frame, it does not have the resolution required to assess whether a frame is late or missing. Since the trunk is bidirectional and operates at the same bit rate, the PARAT should receive DBCES frames into the *vbr_to_cbr process*; and thus produce output CBR 'frames'; at the same rate as the *cbr process* presents CBR 'frames' to the *cbr_to_vbr process*. Thus, providing account is taken of its jitter and the jitter of the received DBCES frame, the reception of a CBR 'frame' can be used to trigger the passing of a CBR 'frame' from the *vbr_to_cbr process* to the *cbr process*.

4.3.3 Buffers

Buffering, with consequent latency accumulation, occurs in the PARAT for both dejitter purposes and because the DBCES processes operate on a frame of channel data. The placement of buffering is shown in Table 2. Note that a Parakeet call will experience this additional end to end latency for each PARAT equipped links traversed in establishing its end to end connection.

Table 2. Parakeet ATM and PARAT Buffering

Location	**Length**	**Length in Time (assuming EUROCOM trunk at 512 kbps)**
Local ATM switch CBR AAL1 frame	47 bytes	0.7 ms
cbr process (in)	32x22 bytes	11 ms
cbr_to_vbr process	~0 bytes	~0 s
vbr process (out)	~0 bytes	~0 s
Local ATM switch VBR real time buffer	4 x 47 bytes (max)	2.8 ms (max)
vbr process (in)	Nx22 bytes	11 ms (max)
vbr_to_cbr process	2x32x22 bytes (max)	22 ms (max)
cbr process (out)	~0 bytes	~0 s
Distant ATM switch CBR buffer to Parakeet switch	4 x 47 bytes (max)	2.8 ms (max)
Total		**50.3 ms (max)**

4.3.4 Action on Lost/Overly Late Traffic

Lost/overly late traffic is unlikely on the CBR service from the local ATM switch to the PARAT concept demonstrator since this link will be only lightly loaded. The PARAT development focussed on loss/overly late traffic on the intersite link (VBR service over SDT AAL1) where errors and ATM switch buffer overflows may sometimes occur. A key concern is the impact on the EUROCOM synchronisation channel. If the trunk group loses synchronisation it will take about 40 ms to re-establish once normal traffic starts flowing again. For short disruptions to the DBCES link, camouflage actions on the synchronisation channel will be required.

4.3.4.1 The vbr process.

The loss of a cell is detected by the *vbr process* through the use of the AAL1 sequence number carried in each cell. The *vbr process* maintains bit synchrony in the SDT AAL1 frame by inserting, for each lost cell, a dummy payload of 47 bytes each with a default value (binary 10101010).

- If the cell comprised only traffic channel data (ie no DBCES frame header) the default value in effect transmits CVSD silence on all channels. In the worst case of the minimum five channels on the DBCES, each channel would experience 9 or 10 bytes (around 5 ms) of silence. For a voice traffic channel this would be barely noticeable, but the loss of this number of bits in the synchronisation channel would cause the system to lose trunk group synchronisation (see later discussion on impact of errors below).

- If the lost cell contained the SDT AAL1 structure pointer, the *vbr process* will typically lose the SDT AAL1 frame boundary. (The only exception is if the lost structure pointer was indicating the frame ended at the end of an eight cell block. In such a case the start of the next frame would have an additional structure pointer.) The *vbr process* has no understanding of the DBCES frame protocol, and can only identify frames via the SDT pointers. As a consequence, aside from the noted exception, the *vbr process* will pass to the *vbr_to_cbr process* a corrupted, frame. The corrupted (concatenated) frame will comprise the first DBCES frame header (with bit mask), uncorrupted data from the first DBCES frame, an ATM cell length of silence pattern, then uncorrupted data from the second frame (but no associated DBCES header).

4.3.4.2 The vbr_to_cbr process.

In the concatenated frame situation, the *vbr_to_cbr process* can still use the DBCES frame header bit mask to extract some channel information. The *vbr_to_cbr process* knows the active channels in the first frame from the bit mask and knows that 22 bytes per channel are available in the frame. Note that up to nearly 5 ms of silence may be incurred on traffic channels and camouflage action will be required on the synchronisation channel. Moreover, depending on the size of the dejitter buffer in the *vbr_to_cbr process*, the first frame may now be too late as the concatenated frame is not received until the end of the second frame (ie the arrival of the third frame). Since the second frame bit mask is absent, no channel data can be extracted from the rest of the frame and camouflage action will be required on the synchronisation channel.

In the PARAT concept demonstrator, the dejitter buffer in the *vbr_to_cbr process* is only two CBR 'frames'. Jitter of the arrival of CBR 'frames' in the *cbr_to_vbr process* will mean that the concatenated frame will sometimes arrive too late to be used. The buffer management algorithm in PARAT concept demonstrator design would discard both frames in this case.

As proposed earlier, the *vbr_to_cbr process* passes a CBR 'frame' to *cbr process* each time a CBR 'frame' is passed by the *cbr process* to the *cbr_to_vbr process*. During initialisation, the PARAT passes zeroed data down the CBR channel and this continues until the *vbr_to_cbr process* has established a stable synchronisation channel in the DBCES frame and a two frame dejitter buffer is established with valid traffic. Synchronisation is accepted as achieved when there are three consecutive bytes of the received synchronisation channel matching a consecutive three byte sequence in the rolling 15 byte synchronisation sequence. Once synchronisation is achieved, the *vbr_to_cbr process* maintains a pointer into the rolling 15 byte synchronisation sequence where the first byte of the synchronisation channel of the next DBCES frame should appear – the synchronisation channel steps forward 22 bytes for each DBCES frame. This pointer is used to check the synchronisation channel in each received DBCES frame to validate that it is in sequence. If the frame is valid the *vbr_to_cbr process* replaces the data in the synchronisation channel with the appropriate segment of the synchronisation sequence before processing the frame further. This camouflages any corruptions in the synchronisation channel caused by errors or cell loss in the *vbr process*. Each time a CBR 'frame' is passed to the *cbr process* that buffer is filled with a dummy frame comprising the appropriate segment of the synchronisation sequence in the synchronisation channel and silence on all other channels. If a DBCES frame is overly late (or lost), then the clocking discipline will result in this dummy frame being sent to the *cbr process* to ensure bit synchrony in the CBR stream and preventing undue Parakeet circuit switch synchronisation loss. Checking of the synchronisation channel using the pointer into the synchronisation sequence will support the identification of overly late frames that can be discarded. If a defined number of dummy frames are sent consecutively (in the case of the PARAT concept demonstrator this is five frames), the *vbr_to_cbr process* declares itself to have lost synchronisation and returns to an initialisation procedure.

5. Silence (Speech Activity) Versus Idle Channel (Call Activity) Detection

Active channels transfer the audio signal via digital patterns created from the analog to digital (A/D) conversion. The A/D standard used by Parakeet is Continuously

Variable Slope Delta Modulation (CVSD) which is a form of adaptive step sized delta modulation. By good fortune, the bit pattern transmitted by the circuit switch on idle channels (continuous "101010" etc) is the bit pattern that results from the CVSD coding of perfect silence. While it is unlikely that an active connection from a busy, noisy command post would be perfectly silent if the handset microphone were in use, the Parakeet telephone handset has a press-to-talk switch to disable the microphone and this can provide perfect silence code to be transmitted.

Everyone has experienced that in normal conversation most speakers are silent for around 50% of the time (allowing the other end to speak). The probability that a voice signal will be silent for a given length of time has been the subject of considerable study [6] and [7] leading up to the ITU standardising a statistical model [8] for voice conversations. This model describes the long term speech activity factor as being 27.6%. Thus a speaker is on average silent for 72.4% of the time – this captures the small silent periods within and between words in addition to the periods when listening. By "silent" the ITU standard means the speech sound is below a given threshold of audio volume. To determine whether a CVSD coded stream is below a threshold effectively requires the stream to be decoded back to analog and this is not feasible in the PARAT demonstrator. However perfect silence can be achieved during the listening period provided the user releases the press-to-talk switch. If one assumes that the periods of silence where the user releases the press-to-talk switch are greater than 2 seconds, then the ITU standard indicates that this would comprise 10% of the total silence period. Thus about 7% bandwidth saving could be achieved by capturing press-to-talk releases. This saving is over and above that that can be made by responding to call (connection) activity.

6. Initial Validation Trials

An initial validation trial was conducted at the Army Communications Training Centre Development Wing. This trial specifically addressed:

- PARAT implementation of the EUROCOM multiplexing standard.
- the correct re-creation of channels in use across the link.
- the transparent transmission of the trunk.
- calculation of bandwidth consumption for different channels in use.
- the impact on data throughput of released bandwidth.

The testing involved connecting two Parakeet switches via the PARAT device. Its purpose was to verify the correct operation of the PARAT in various use scenarios such as X.25 data connections and high usage.

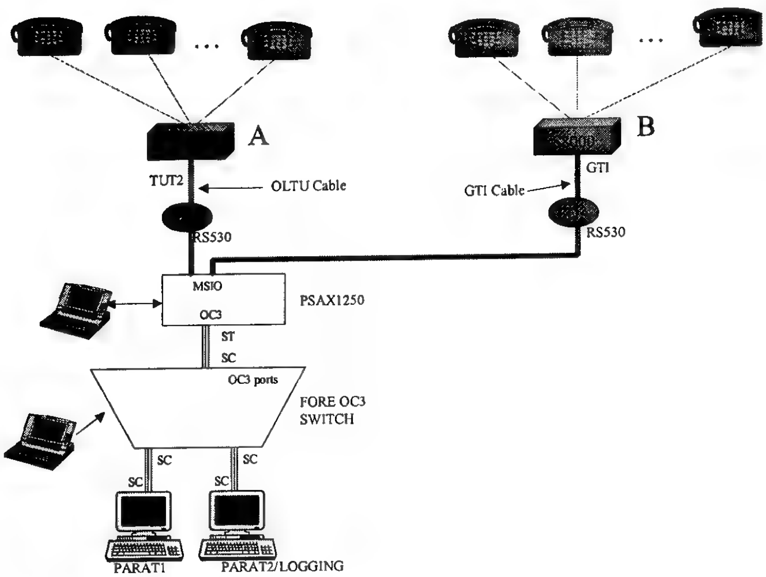

Figure 5- PARAT trial configuration

Figure 5 shows the configuration used during the trial to test the functionality of the PARAT. The two Parakeet switches were connected to the PSAX1250 ATM switch, which then connected to the FORE ATM switch. This was then connected to the two PARAT machines. Using the above set up it was possible to test various interconnection scenarios such as running a loop back trunk over the PSAX1250. (Note that in operational use, there would be two PSAX1250s connected via a satellite or radio relay trunk link.

The testing validated the operation of the PARAT, which performed well for both the 512kbps and 256kbps trunk. A rough test of the bandwidth saving made by the PARAT can be seen below.

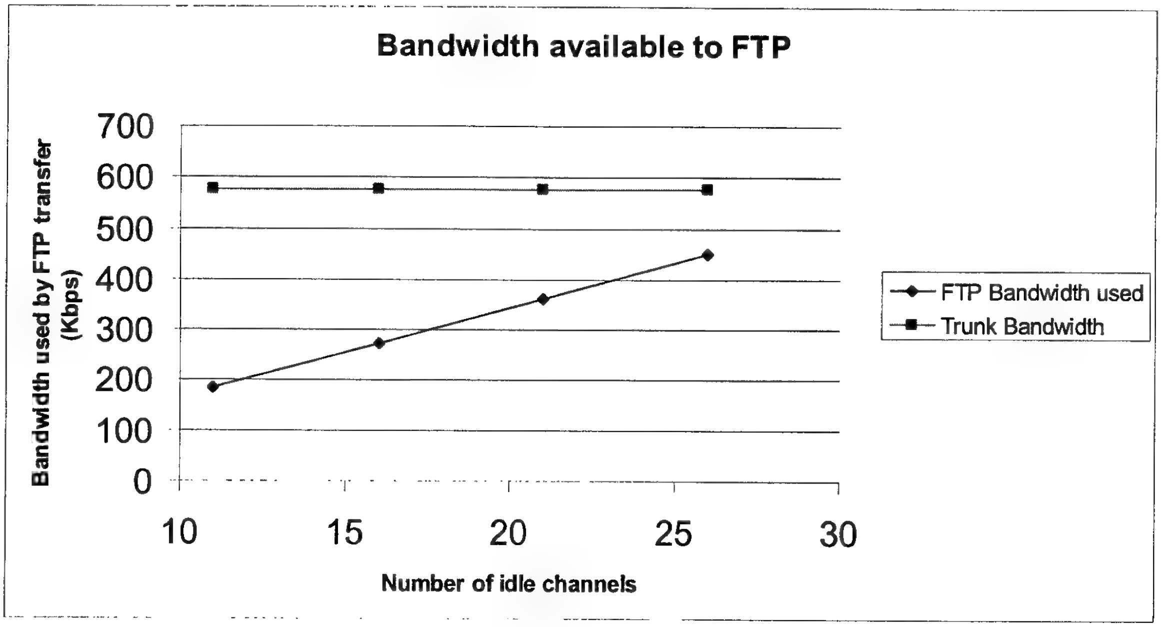

Figure 6 – Bandwidth available to FTP when using PARAT

The graph in Figure 6 shows the bandwidth used by a single FTP session when the link was shared by the PARAT. The trunk used in this test had an aggregate 576kbps rate, with the PARAT DBCES cell stream and a HLDC router link carried across the trunk. As can be seen in the graph, as the activity in the trunk drops the bandwidth available (and used) by the FTP session increases. There was an interesting result shown up by this testing, and it is the fact that the projected available bandwidth freed up by the PARAT did not match the bandwidth used by the FTP session (a single TCP/IP data connection). The two values differ by about 10kbps, and this can be explained by the way a TCP connection limits its bandwidth usage. The back-off algorithm used by TCP interacts with the bursty DBCES cell transmissions and causes the bandwidth used by the TCP connection to fluctuate. Nevertheless, the slight difference is negligible.

The trial did not reveal any major issues with the PARAT concept or the PARAT implementation, with more than 4.8 billion cells being processed by the PARAT. The trial tested the various components of the system verifying: the correct operation of the AAL1 stack, the SDT mode, the DBCES frame handling and the various EUROCOM operations. Due to there being tests with more that 17 active channels the SDT stack was also verified for operation when the DBCES frame size exceeded 8 ATM cells (see Appendix B for more details).

The only remaining test for the PARAT device is to run it over a satellite connection to check that there are no latency problems, but this is not seen as an issue.

7. Further Work

7.1 Operational Configuration

We believe that the PARAT concept demonstrator with its adjunct processor approach may not be appropriate as a fully fielded system, principally because of concerns about scaleability. Definitive testing has not been possible to determine how may simultaneous trunks can be handled by the workstation PC but as more trunks are added, the performance challenge on the PC increases more than linearly. The requirement for the DAHA is three trunks - the requirement for the satellite hub station will be larger still - and this may exceed the capability of adjunct processor based solution. Nevertheless, a system based on a COTS embedded PC can provide the small footprint and robustness required yet keep the development costs low because they can leverage off the concept demonstrator.

A more scaleable approach to the design of an operational PARAT capability would be to have devices each operating on a single trunk in line between the circuit switch and the ATM switch. An implementation approach might be to increase the functionality of the current EDA from the present simple physical layer adaption and turn it into a EUROCOM to DBCES/ATM adaptor. This approach would also lend itself to digital signal processing/programmable gate array technology that is more capable in implementing error protection mechanisms. However, such an approach will incur significant non-recurring engineering costs as the PARAT process is reasonably complex with a large amount of sub-systems. With the limited number of ATM switches being fielded and the expected lifetime of the Parakeet switches, it may be more economical to settle for the mid ground in the operational device.

The final decision about which direction to take should be based on an analysis of the number of units that will be fielded and the concept of operations for the device (with the inline card being the optimal engineering choice, and the adjunct processor PARAT being the cheaper solution).

7.2 Impact of Errors

The Parakeet specification [5] calls for the system, including signalling, to operate over transmission bearers with $1x10^{-4}$ bit error rate. Frame alignment is required to be able to be recovered within 40 ms with a probability of at least 50% in error rates of $1x10^{-2}$. It should be noted that while the EUROCOM standard [1] requires the system remain in synchronisation even at error rates of 10%, this is averaged over 5 ms. For the suggested synchronisation circuitry described in the standard, a burst of errors (in the order of 60 bits) would result in loss of synchronisation.

The PARAT communications link moves information in a structured manner. The impact on each of the different structural elements from bit errors on the link is discussed below. The calculations assume random bit errors at 1x10^{-4} bit error rate.

- The ATM header. See an earlier paper by Wilksch [9]. Errors in the header can result in cell delineation loss (very unlikely even at this error rate) or cell redirection (loss) of about 4%. This rate is unacceptable, but the particular ATM switch chosen by Parakeet offers an "error tolerant addressing" mode discussed in [10] which would give in order of 10^{-7} cell loss or around three hours between cell loss from this fault mechanism.
- AAL1 header. The AAL1 header has a checksum that can detect bit error requiring the cell to be discarded - hence an impact the same as cell loss discussed earlier. In a bit stream with a random BER of e_b (where e_b is a value between 0 and 1), the probability that a particular bit is not in error is 1 - e_b. The probability that n selected bits from that stream are all correct is $(1 - e_b)^n$. Hence, the chance that there is at least 1 error in those n bits is $1 - (1 - e_b)^n$. Thus the chance of an error in AAL1 Header (8 bits) is $1 - (1 - e_b)^8$ or 8x10^{-4}. When all channels are active, PARAT produces 1.4x10^3 cells per second thus AAL1 header error would occur on average every second. 94% of these would impact directly on the voice traffic (an imperceptible 1.5 ms of silence) the remainder would result in frame loss (22 ms of silence). When running on minimum channels active, PARAT produces 222 cells per second thus AAL1 header error would occur on average every 5.6 s with half resulting in frame loss and half impacting directly on the voice traffic with 5 ms of silence inserted.
- SDT pointer. An error in the SDT pointer will result in a loss of DBCES frame synchronisation as discussed earlier. The chance of an error in SDT pointer (8 bits) is $1 - (1 - e_b)^8$ or 8x10^{-4}. Since the frame occurs each 11 ms, a DBCES frame would be lost on average each 13.8s.
- DBCES frame header. An error in the frame header will typically result in a mismatch between the indicated active channels and those actually being carried. This may result in a loss of traffic for the period of the frame. The chance of an error in DBCES frame (32 bits) is $1 - (1 - e_b)^{32}$ or 3.2x10^{-3}. Since the frame occurs each 11 ms, disruption on average each 3.4 s.
- DBCES payload. This will result in errors (noise) in the traffic itself. CVSD quite resilient, certainly capable of coping with the likely error rates

In cases where the channel has a forward error correction coding (FEC) capability, errors might not be random, but instead occur in bursts. For instance typical satellite modems produce burst of 6 bit errors. Calculations of the impact of such burst errors can employ the same formula as above, with the figure for e_b divided by the number of bits in the burst and the figure for n increased by number of bits in the burst minus 1. In general bursty errors will lead to better performing systems than with random errors, for instance the DBCES frame header errors would occur on average 17.8 s.

It should be noted that errors to the ATM header, AAL1 header and traffic payload apply equally to the current ATM implementation on Parakeet as well as PARAT. The additional PARAT vulnerabilities relate to the SDT mode and the DBCES frame header. At a very small price in overhead, for instance perhaps an additional four bytes for the four byte DBCES header, error rates could be reduced effectively to zero. Selective FEC, ie on the important structural elements rather than the entire stream, is however a slow process in a software processor system such as the PARAT concept demonstrator. A hardware based PARAT using DSP or FPGA technology, being bit oriented, could implement FEC and maintain real time performance.

7.3 Non Constant Processing Overhead

The amount of processing that is required for a trunk is inversely proportional to the number of active channels. This means a lightly used trunk will require more processing that a busy one. This non-constant processing overhead means the number of trunks that a single PC can handle varies depending on the activity of the trunk.

Another non-constant processing overhead occurs in the re-assembly of a corrupt AAL1 stream. If an extremely corrupt stream (90% cell loss or above) is passed to the PARAT the processing load dramatically increases because of the large amount of cell reordering that must be done to reconstitute a valid AAL1 stream.

Any scaling of the adjunct processor concept must be done after careful consideration of the extreme conditions that could be experienced by the device.

7.4 Line Rate Detection

The current PARAT implementation requires the user of the program to input the trunk line rate that is being used. With some further work, it should be possible to automate this process. There are two methods that could be used to detect the line rate. The first is a simple rate check. Even though the timing from the machine is reasonably coarse, by dividing the time to receive a cell by its size it should be possible to determine the line rate. Another method involves reading the cell stream in as a 512kbps stream by default. If two synchronisation channels are detected in the de-multiplexed stream then it is more than likely that a 256kbps trunk is being read. The only caveat with both methods is that the line rate must be recalculated when the synchronisation channel is lost (to account for the when a new line speed is being engineered).

8. Conclusion

In summary:

- Concept of a EUROCOM VBR ATM trunk has been proven.

- Mechanism for idle channel identification have been proven, plus the added benefit that idle channel looking like silent channel has been revealed.

- The benefits in bandwidth reclamation employ two separate mechanisms:
    - Call activity – ie savings because entire 30 channel trunk is not allocated to interswitch calls
    - Speech activity. This itself comprises two elements:
        - The natural silence when user is listening to other end – PARAT can take advantage of these silences provided users release the press to talk switch
        - Small silent periods within the speech spurt (pauses between words and phonemes) – PARAT may not capture these silent periods

- The PARAT concept demonstrator could perhaps become basis of an initial operational capability, buta scalable solution may require further development by industry

- Some issues such as greater error resilience may be subjects of further consideration.

9. References

[1] "Tactical Communications Systems Basic Parameters", EUROCOM D/1.

[2] W.D. Blair, "A Case for Call Activity Detection for ATM Voice Services in Project Parakeet," DSTO Report DSTO-TR-0606, 1998.

[3] ATM Forum Technical Committee Baseline Document atm96-0136R, "Specification of Dynamic Bandwidth Utilization in 64 Kbps Time Slot Trunking over ATM Using CES." 14 November 1996.

[4] ITU-T Recommendation. I.363.1, "B-ISDN ATM Adaptation Layer specification: Type 1 AAL", 1996.

[5] Commonwealth of Australia "JPB65- Project Parakeet Tactical Trunk System Phase 4.1/4.2. Attachment H, System Specification, Volume Three", 1994.

[6] P.T. Brady, "A Statistical Analysis of On-Off Patterns in 16 Conversations," *Bell Syst. Tech. J.*, vol. 47, no. 1, pp. 73-91, Jan. 1968.

[7] H. H. Lee and C. K. Un, "A Study of On-Off Characteristics of Conversational Speech," *IEEE Trans. Commun.*, vol. COM-34, no. 6, pp. 630-637, Jun 1986.

[8] ITU-T Recommendation P.59 "Artificial Conversational Speech", 1993.

[9] D.M. Wilksch, "Project Parakeet – The Performance of Asynchronous Transfer Mode over Network Trunks," DSTO Report DSTO-TR-0551, 1997.

[10] D.M. Wilksch and L. Dong, "Measured Performance of the Comsat Asynchronous Transfer Mode Link Accelerator – Channel Noise and Interference Tests," DSTO Report DSTO-TR-0860, 1999.

Appendix A: EUROCOM Trunk Characteristics

A.1 EUROCOM TDM Frame Structure

Parakeet uses either 512 kbps (30 traffic channels) or 256 kbps (14 traffic channels) Time Division Multiplex (TDM) trunks as defined in EUROCOM [1]. Two time slots are dedicated for each of Frame Synchronisation and Common Channel Signalling. Individual traffic channels use 16 kbps Continuously Variable Slope Delta (CVSD) coding. The TDM stream comprises of the synchronisation channel, the signalling channel and the relevant number of traffic channels, bit interleaved (ie one bit per channel per frame). The framing structure for 512kbps is given in Figure A1 and is sent onto the line from left to right.

Tfc29	Tfc30	Synch	Signa1	Tfc1	Tfc2	Tfc3	Tfc4	Tfc5	Tfc6	Tfc7	Tfc8	Tfc9	Tfc10	Tfc11	Tfc12	Tfc13	Tfc14	Tfc15	Tfc16	Tfc17	Tfc18	Tfc19	Tfc20	Tfc21	Tfc22	Tfc23	Tfc24	Tfc25	Tfc26	Tfc27	Tfc28	Tfc29	Tfc30	Synch	Signal	Tfc1	Tfc2

Figure A-1. EUROCOM TDM frame format.

A.2 EUROCOM Synchronisation

The synchronisation channel carries a continuously repeating 15 bit pattern (transmitted in order from left to right):
...01/000011101100101/00...

If the local switch is in alarm (for instance if it has not synchronised to the incoming synchronisation channel) it transmits a "housekeeping" pattern comprising an inverted synchronisation sequence:
...10/11100010011010/11...

Since the PARAT captures and processes individual channels (including the synchronisation channel) in bytes, and the 15 bits result in a shifting byte boundary when segmented, this leads to a continuously repeating 15 bytes pattern apparent in a byte oriented channel stream. This is shown in Table A1 and A2. (Note that the bytes are filled from the bit stream in order from most significant bit to least significant bit in accordance with ATM standards).

10000111 = 135
01100101 = 101
00001110 = 14
11001010 = 202
00011101 = 29
10010100 = 148
00111011 = 59
00101000 = 40
01110110 = 118
01010000 = 80
11101100 = 236
10100001 = 161
11011001 = 217
01000011 = 67
10110010 = 178
10000111 = 135
01100101 = 101
00001110 = 14
11001010 = 202

Table A-1. Byte Oriented Synchronisation Stream (with Decimal Equivalent)

01111000 = 120
10011010 = 154
11110001 = 241
00110101 = 53
11100010 = 226
01101011 = 107
11000100 = 196
11010111 = 215
10001001 = 137
10101111 = 175
00010011 = 19
01011110 = 94
00100110 = 38
10111100 = 188
01001101 = 77
01111000 = 120
10011010 = 154
11110001 = 241
00110101 = 53

Table A-2. Byte Oriented Local Alarm Synchronisation Stream (with Decimal Equivalent)

Synchronisation is accepted as achieved when processing achieves three consecutive bytes of the synchronisation channel matching a consecutive three byte sequence in the rolling 15 byte synchronisation sequence. The probability of falsely achieving synchronisation on a random channel is 2^{-24} or approximately 10^{-17}.

A.3 Idle and Silent Channel Identification

CVSD is fundamentally an adaptive step-size delta modulation. When the channel is silent, ie a connection has been successfully made, but no audio is being exchanged (perfect silence), the CVSD coding produces a continuing repeating bit pattern of '10'. When collected into a byte this pattern is equivalent to decimal 170 or 86 depending on byte alignment (ie whether the byte collected is '10101010' or '01010101'). In the case of Parakeet where the handset has a pressel switch, releasing of this switch removes any local background noise and creates a perfectly silent source.

When a channel is not connected, it is considered by the Parakeet Circuit Switch to be idle. The bit pattern to indicate to the distant switch that the circuit is idle is not defined in the EUROCOM standards. As it happens, the distinctive pattern placed on an idle traffic channel by the Parakeet Circuit Switch (Tadiran TD8000 or Tadiran TD8500) is identical to the CVSD silent pattern.

Appendix B: Design Criteria for PARAT DBCES PDU

B.1 DBCES PDU Structure

The commercial standard DBCES protocol is carried via ATM AAL1 in a Structured Data Transfer (SDT) mode. In such a mode, there are two elements of overhead in addition to the normal ATM five byte header overhead.

- In conventional AAL1 one byte of the 48 byte ATM payload is set aside to carry a sequence number. Each AAL1 ATM cell is sequence numbered from 0 to 7. The sequence number can be used to identify the loss of cells so that dummy ATM cells can be inserted at the receiving end to maintain bit synchrony. SDT also uses a bit within the AAL1 byte to indicate whether the cell contains a second byte of SDT specific overhead.
- Periodically, SDT adds this second byte of overhead. This byte acts as a pointer (or offset value) to the start of a data structure within the current or next ATM cell. The data structure is segmented into the payload area of ATM cells. Unlike some other ATM modes, there is no padding of dummy bytes to ensure each element of structured data occupies an integral number of ATM cells; instead subsequent structures are concatenated as a continuous stream of structures. Thus data structures can commence at any point in an ATM cell and hence the need for the pointer (especially to re-establish structure synchrony in the case of cell loss). This pointer must be sent on an even sequence number and standards recommend that it be sent not less often than every eight cells. In the event that a new structure does not commence in an eight cell sequence (specifically from a sequence number 0 to the subsequent sequence number 7), a dummy SDT pointer is placed in the ATM cell which has the sequence number 6. If, by chance the structure ends exactly on the end of that run of eight cells, the dummy value is 93; if the structure continues past the end of the eight cells, the dummy value is 127.

The commercial DBCES PDU comes in two forms:

- A Type 1 PDU comprises a four byte bit mask where each bit signifies whether a channel in the original TDM trunk is active (and hence being carried) followed by one byte of voice data per active traffic channel in channel order. The SDT pointer points to the first byte of the four byte bit mask.
- A Type 2 PDU comprises just the traffic – one byte per active traffic channel in channel order.

For commercial Pulse Code Modulation systems, each sample period produces an eight bit (one byte) figure representing signal amplitude at the sample time. Thus in the DBCES, the PDU carries one sample value for each traffic channel. For Parakeet the traffic channels are bit oriented. Each sampling period in the original analog to digital conversion produces a single bit. Nevertheless, it is more convenient to process channels using bytes of traffic. Thus, when the DBCES PDU definition is adapted for

PARAT, each PDU holds eight time slots of data for each traffic channel. Since each traffic channel is operating at 16 kbps, each PDU holds 0.5 ms of traffic for each active channel.

The PARAT collects 22 bytes of traffic data per channel for processing. The DBCES frame (SDT structure) would thus comprise four bytes of bit mask followed by 22 bytes of traffic data for each active channel interleaved one byte per channel. (Consideration might be given in future PARAT developments to using a two byte bit mask when operating in the 16 channel/256kbps mode where two bytes are sufficient for the 16 channels.) In effect this is one Type 1 PDU and 21 of Type 2. However, if there were four or fewer channels active, the resultant DBCES frame would be less than two full ATM cells. This would result in a situation where there are two frames starting in the same consecutive pair of ATM cells, but the SDT pointer can only point to one bit mask in each pair of cells. Accordingly, PARAT will always send a minimum of five channels even though some may be carrying silence/idle patterns. With more than 15 channels of active data the SDT pointer is used in the extended frame mode when the SDT pointer value is either 93 or 127 depending on when the DBCES frame finished in the cell stream.

While there is some penalty for putting this lower limit on channels carried, this is not overly burdensome. Also, there is a useful side benefit. Consider the case of a single channel active (a DBCES frame of 4 byte bit mask and 22 bytes of data) and ignore the limitation of SDT pointers only being able to point to one structure. Since, in AAL1, ATM cells cannot be sent until filled, one would require more than two frames to fill a cell. This would mean that there would be additional delay at the sending end before a frame is sent - this would likely result in buffer underflow at the distant end (or a requirement for longer dejitter buffers with consequent increase in end to end delay).

Appendix C: Outline Software Design for PARAT

C.1 Architecture

The software design for the PARAT implementation is governed by the concept of modularity and the OSI protocol stack model. To achieve this, the various areas of functionality are implemented in various source code files, with each file implementing a layer of the protocol stack. Due to the tight coupling of ATM AALs to the hardware, this rule is broken occasionally but it is documented and the interfaces between the layers are kept as simple as possible.

C.1.1 Source Code Layout

The source code is broken up among the following files.

main.c - the code in this file is the glue for the system. It contains the logic to implement the high level PARAT concepts, and serves as the main entry point into the program.

dbces.c - this implements the concepts of DBCES. It contains functions to send and receive dbces frames. It can convert from a DBCES from to an uncompressed frame, and vice versa. The dbces_recv() function is independent of the ATM read/write functions, while the dbces_send() function calls struct_aal1_send() to send the frame it creates.

struct_aal1.c - Structured AAL1 sends and receives are implemented in this file. It implements the construction of frames of data, and the reassembly of frames received from the lower AAL1 calls.

aal1.c - The Segmentation and Reassembly operations of AAL1 are performed by the functions in this file. It also incorporates some structure pointer operations due to the coupled nature of structure pointers and AAL1.

atm_funcs.c - the lowest level ATM API functions are contained within this file. It has functions to read and write ATM cells to the ATM NIC. If this program is to be ported to another platform these functions should be changed to reflect the new platform.

demux.c - this file contains the various functions to de-multiplex and re-multiplex the EUROCOM TDM structure. It also contains functions to detect the sync signal of a trunk in its de-multiplexed state.

buffer.c - the buffering structure implemented by the PARAT is in this file, with various support functions.

C.1.2 Division of Labour

The PARAT program itself is broken into three threads. One thread handles reading in of CBR cells, one reads in VBR cells and the main thread performs the multiplexing/de-multiplexing and sending of the VBR and CBR streams. This grouping differs slightly from the architecture presented in this paper because of the blocking nature of "read" system calls. The CBR and VBR input streams block until a complete frame of data is received. This data is then sent to the main process via pipes so it can process the information and produce output cells.

C.1.3 Interesting Features

The software has a couple of novel concepts that will now be discussed.

Firstly, the buffering code is implemented by an array of 15 separate buffers, with each buffer containing a dummy frame and a position for a valid frame. The 15 separate entries map to the 15 different byte positions possible for the first byte in the sync channel. The VBR process inserts valid cells into the position that maps to the first byte in the decoded VBR cell. The CBR process steps through the array in numerical order, writing out valid frames where they exist and sending dummy frames when needed. This circular array minimised processing overhead when valid frames are missing or corrupt because the correct dummy frame has been pre-computed, so a lost frame does not result in a processing hit. The dummy frame that is written depends on weather the incoming stream is in an alarm condition (inverted synchronisation channel). If it is alarmed an inverted synchronisation channel buffer is used, otherwise a normal pattern is used. The reason for using this inverted pattern is to provide the correct signals to the near switch. If the normal pattern was always sent then a race condition happens when the two switches continually drop in and out of synchronisation.

The second novel idea is the way the ATM cell stream is represented. Instead of storing an array of separate cells, and carefully stepping through them to perform reassembly, the cells data is stored as a simple linear character array. The ATM and AAL1 headers are stripped off the cell, checked for errors and examined for the Structure Pointer. If a structure pointer exists, a linked list of current structure pointers is added to, the SP byte stripped and the remaining data is copied into the array. Segmentation is performed in roughly the reserve order.

C.1.4 Issues

There is an issue with the multiplexing of the bit based EUROCOM trunk on the byte based x86 architecture. Quite a bit of processing power is devoted to merely to pulling apart and reconstructing the trunk. This issue could easily be solved using a FPGA solution however due to the bit based processing that could be performed. To improve performance an assembly version of the de-multiplex routine has been created using the MMX instruction set to take advantage of the SIMD (Single Instruction Multiple Data) parallelism. This MMX implementation provided a three-fold improvement over the simple 'C' code version.

DISTRIBUTION LIST

Implementation of Call Activity Detection for ATM Voice Services in Project Parakeet

A.B. Reynolds and W.D. Blair

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor **DGC4**

S&T Program
Chief Defence Scientist } shared copy
FAS Science Policy } shared copy
AS Science Corporate Management } shared copy
Director General Science Policy Development } shared copy
Counsellor Defence Science, London (Doc Data Sheet)
Counsellor Defence Science, Washington (Doc Data Sheet)
Scientific Adviser to MRDC Thailand (Doc Data Sheet)
Scientific Adviser Policy and Command
Navy Scientific Adviser (Doc Data Sheet and distribution list only)

Scientific Adviser - Army
Air Force Scientific Adviser
Director Trials

Aeronautical and Maritime Research Laboratory
Director

Electronics and Surveillance Research Laboratory
Director (Doc Data Sheet and distribution list only)

Chief of Communications Division
Research Leader, MIN Branch
Head Network Architecture Group
Head Network Management Group
Head Wireless Systems Group
W.D. Blair
A.B. Reynolds
J.T. Ball

Chief of Land Operations Division
Research Leader, LS Branch
F. Bowden

DSTO Library and Archives

Library Fishermans Bend (Doc Data Sheet)
Library Maribyrnong (Doc Data Sheet)
Library Salisbury
Australian Archives
Library, MOD, Pyrmont (Doc Data sheet only)

US Defense Technical Information Center, 2 copies
UK Defence Research Information Centre, 2 copies
Canada Defence Scientific Information Service, 1 copy
NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

Capability Systems Staff

Director General Maritime Development (Doc Data Sheet only)
Director General Land Development
Director General Aerospace Development (Doc Data Sheet only)

Knowledge Staff

Director General Command, Control, Communications and Computers (DGC4) (Doc Data Sheet only)
Director General Intelligence, Surveillance, Reconnaissance, and Electronic Warfare (DGISREW)R1-3-A142 CANBERRA ACT 2600 (Doc Data Sheet only)
Director General Defence Knowledge Improvement Team (DGDKNIT) R1-5-A165, CANBERRA ACT 2600 (Doc Data Sheet only)

Army

Stuart Schnaars, ABCA Standardisation Officer, Tobruck Barracks, Puckapunyal, 3662 (4 copies)
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO Gallipoli Barracks, Enoggera QLD 4052 (Doc Data Sheet only)
Army Communications Training Centre
Headquarters
Development Wing

Intelligence Program

DGSTA Defence Intelligence Organisation
Manager, Information Centre, Defence Intelligence Organisation

Acquisitions Program

PD.Parakeet
PD BCSS
PD DJFHQ (Afloat)
PD JP2072
Land Engineering Agency (2 Copies: Library and G. Lampard)

Corporate Support Program

Library Manager, DLS-Canberra

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy
  Library
  Head of Aerospace and Mechanical Engineering
Serials Section (M list), Deakin University Library, Geelong, 3217
Hargrave Library, Monash University (Doc Data Sheet only)
Librarian, Flinders University

OTHER ORGANISATIONS

NASA (Canberra)
AusInfo
State Library of South Australia
Parliamentary Library, South Australia

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS

Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

INFORMATION EXCHANGE AGREEMENT PARTNERS

Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US

SPARES (5 copies)

Total number of copies: 62

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)
nil

2. TITLE Implementation of Call Activity Detection for ATM Voice Services in Project Parakeet	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)
4. AUTHOR(S) W.D. Blair and A.B. Reynolds	5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108 Australia

6a. DSTO NUMBER DSTO-TR-1149	6b. AR NUMBER AR-011-861	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE November 2000

8. FILE NUMBER E8730/15/15	9. TASK NUMBER JNT99/141	10. TASK SPONSOR DGC4	11. NO. OF PAGES 30	12. NO. OF REFERENCES 10

13. URL ON WORLDWIDE WEB http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-1149.pdf	14. RELEASE AUTHORITY Chief, Communications Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Military communication, Project Parakeet, asynchronous transfer mode, time division multiplexing, voice communication, communications networks

19. ABSTRACT

Asynchronous Transfer Mode (ATM) has been introduced into the land tactical communications system (specifically the Parakeet system) and offers the potential for increasing the effectiveness of the communications trunks by dynamically sharing the capacity between competing demands. Such a capability has been limited in respect of the trunks between the Parakeet circuit switches since Parakeet uses military standard protocols that do not integrate easily with the civil standard ATM. The Parakeet Adaptive Rate ATM Trunk (PARAT) was conceived to overcome this interface problem. The PARAT offers substantially improved data communications performance where lulls in voice usage can be exploited by data services sharing the ATM link. This report describes PARAT, in particular the mechanism to identify the moment-to-moment capacity requirements of the Parakeet inter-switch trunk and the method of carrying the variable bit rate stream over ATM.